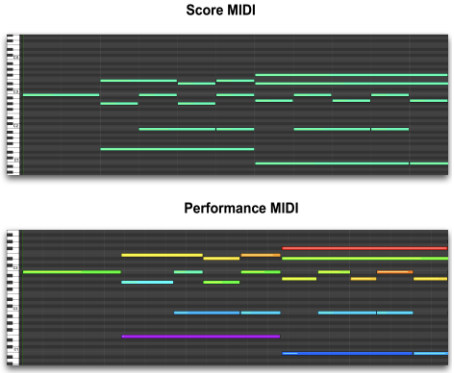
GCT634/AI613: Musical Applications of Machine Learning

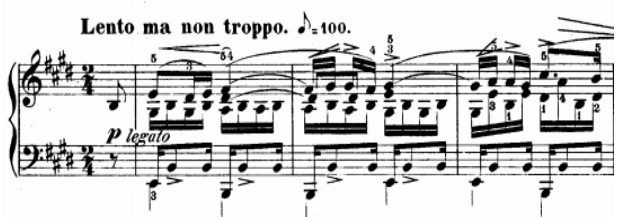
Symbolic Music Generation: Basics

Juhan Nam

Overview of Symbolic Music Generation

- Symbolic music formats
  - MIDI, typesetting (e.g. musicXML), text-based notations (e.g. ABC)
- Tasks
  - Melody-to-Melody or Song-to-Song: continuation, interpolation, infilling
  - Melody-to-{Song, Chord, Lyrics, …}: harmonization/arrangement
  - Score-to-Performance MIDI: performance rendering

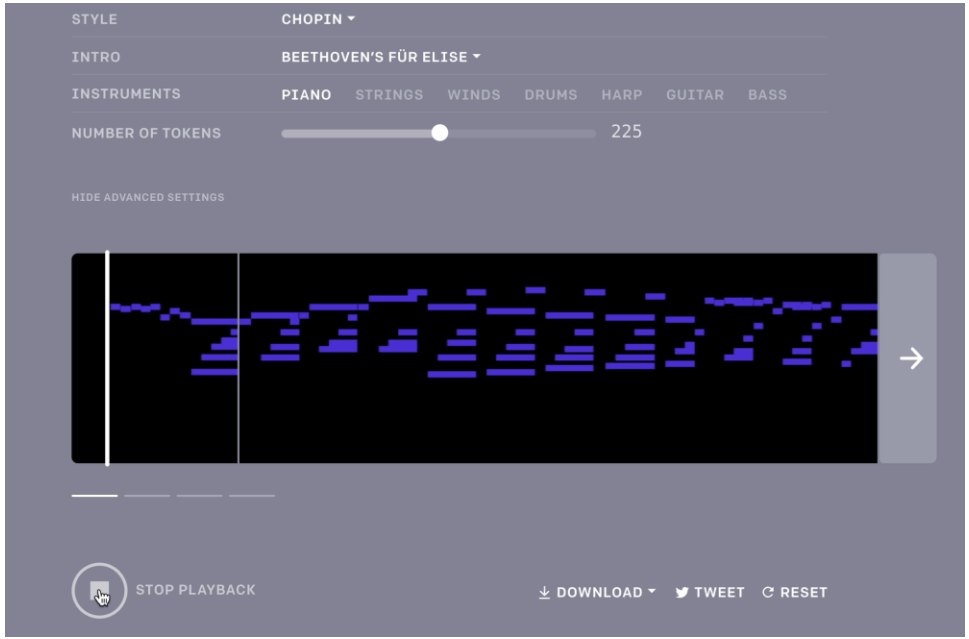





```
<score lang="ABC">
X:1
T:The Legacy Jig
M:6/8
L:1/8
R:jig
K:G
GFG BAB | gfg gab | GFG BAB | d2A AFD |
GFG BAB | gfg gab | age edB |1 dBA AFD :|2 dBA ABd |:
efe edB | dBA ABd | efe edB | gdB ABd |
efe edB | d2d def | gfe edB |1 dBA ABd :|2 dBA AFD |]
</score>
```

MuseNet

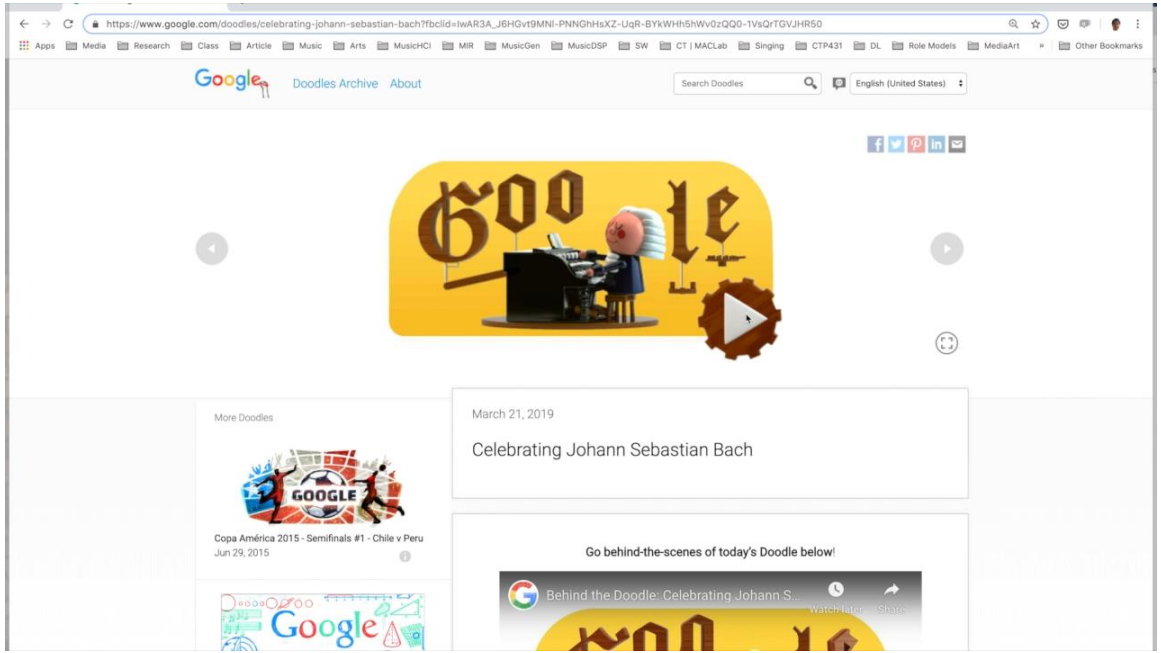
- Song continuation



https://openai.com/index/musenet/

CoCoNet

- Harmonization



https://magenta.tensorflow.org/coconet

VirtuosoNet (2019)

- Music XML → Performance MIDI

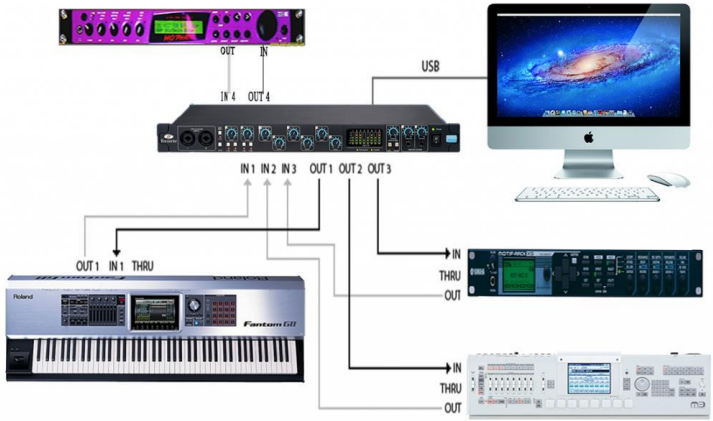
Overview of Symbolic Music Generation

- 1D Models
  - Represent symbolic music data as **discretized token sequences**
  - Train a **language model** using the 1D token sequences
  - Generate new tokens from the trained model in an auto-regressive way

- 2D Models
  - Represent symbolic music data as **continuous image streams** (piano-roll)
  - Train an **image generation model** using the 2D image
  - Generate new image outputs from the trained model chunk by chunk

MIDI

- Standard protocol of musical events
- Why MIDI?
  - Need of musical communication among different vendors' instruments
  - Store music events (score or performance) for composers
- Hardware
  - 5-pin cables, separate in/out in connection
  - 31250 bits per second
- Software (Protocols)
  - Note number/velocity, control data



MIDI

- MIDI Message Format

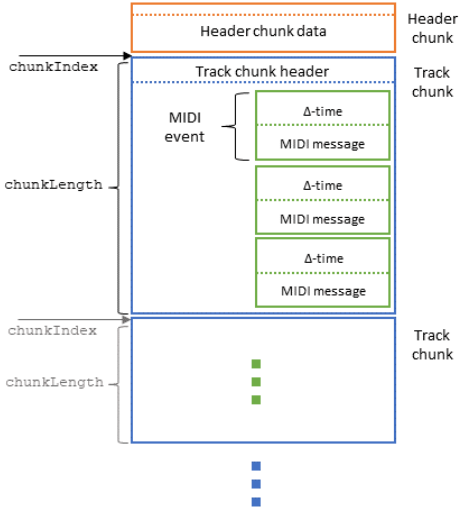
	Status Byte	Data Byte1	Data Byte2	
Note Off	1000 xxxx	Note Number	Velocity	
Note On	1001 xxxx	Note Number	Velocity	
Note Pressure	1010 xxxx	Note Number	Velocity	
Control Change	1011 xxxx	Ctrl. Number	Ctrl Value	→ 64: piano sustain pedal
Program Change	1100 xxxx	Prog. Number	-	
Pitch Bend Change	1110 xxxx	Value (high 7bits)	Value (low 7bits)	

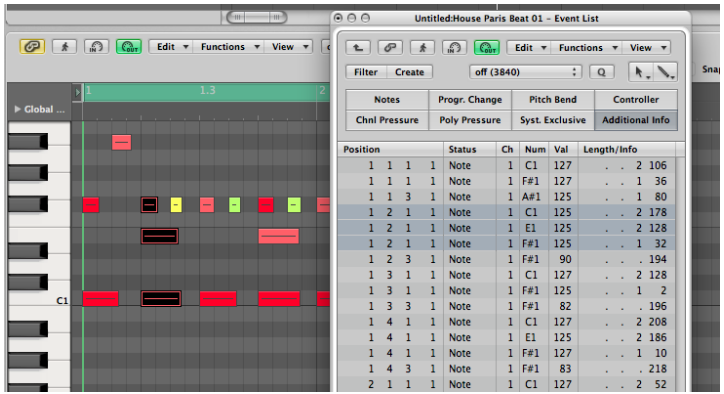
xxxx: channel number (0-15)

Data byte: 0-127 (MSB is 0)

MIDI as a File (.mid)

- Time signature (e.g. 4/4) and tempo (120 bpm) added
  - Time unit is changed to measure/beat unit
  - Time resolution: “tick” (e.g. 9600 ticks/beat)
  - A time interval from the previous event is added to each MIDI message
- MIDI Representations
  - 1D sequence of note events
  - 2D piano-roll images
- “Score MIDI” vs. “Performance MIDI”
  - The beat unit is not meaningful in Performance MIDI



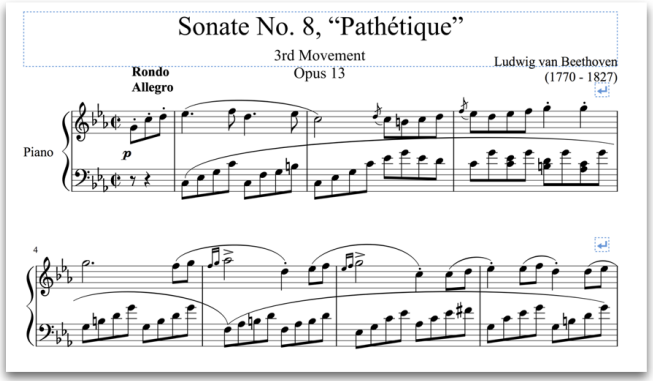


Source: https://kr.mathworks.com/help/audio/ug/convert-midi-files-into-midi-messages.html

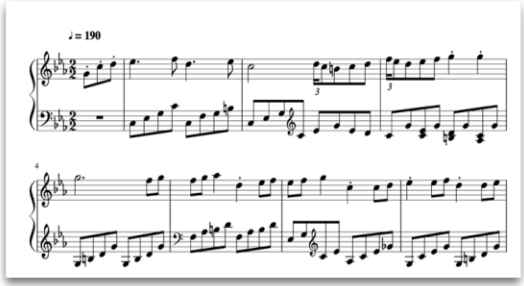
MusicXML

- Markup language for music typesetting (or engraving)
  - Aim to render a realistic music score

```
<part id="P1">
  <measure number="1">
    <attributes>
      <divisions>1</divisions>
      <key>
        <fifths>0</fifths>
      </key>
      <time>
        <beats>4</beats>
        <beat-type>4</beat-type>
      </time>
      <clef>
        <sign>G</sign>
        <line>2</line>
      </clef>
    </attributes>
    <note>
      <pitch>
        <step>C</step>
        <octave>4</octave>
      </pitch>
      <duration>4</duration>
      <type>whole</type>
    </note>
  </measure>
</part>
```



MIDI

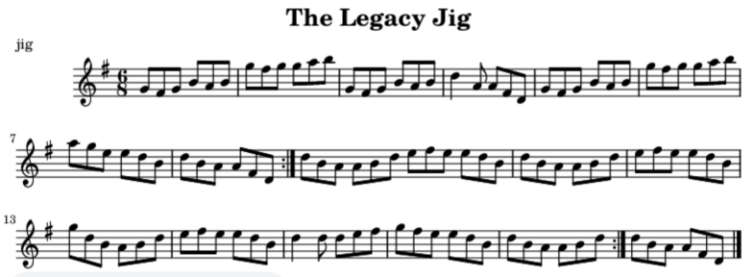


- Can be converted to MIDI but lose information (expressions, clef, articulation)
- There are other music typesetting formats: Lilypond, MEI, …

ABC Notation

- A simplified musical notation format
  - Focus on monophonic melody

```
<score lang="ABC">
X:1
T:The Legacy Jig
M:6/8
L:1/8
R:jig
K:G
GFG BAB | gfg gab | GFG BAB | d2A AFD |
GFG BAB | gfg gab | age edB |1 dBA AFD :|2 dBA ABd |:
efe edB | dBA ABd | efe edB | gdB ABd |
efe edB | d2d def | gfe edB |1 dBA ABd :|2 dBA AFD |]
</score>
```



- There are other simplified music notation formats
  - Humdrum (kern representation), JAM notation (chord), Music Macro Language (game music)

https://en.wikipedia.org/wiki/ABC_notation

1D Models

- Language model in natural language processing
  - Auto-regressive model: predict what comes next

The sky is so _____

x_1 x_2 x_3 x_4 …

blue

beautiful

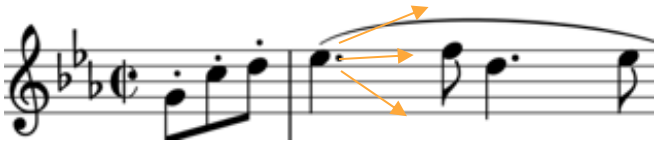
dark

Language Model

$$p(x_t | x_1, \ldots, x_{t-1})$$

x_i: input/output representation vector

- Musical language model
  - Predict what comes next in a note sequence



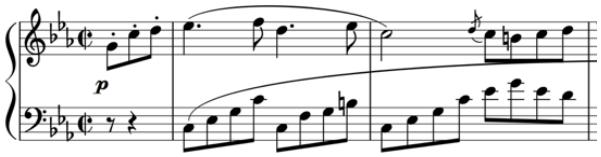
x_1 x_2 x_3 x_4 …

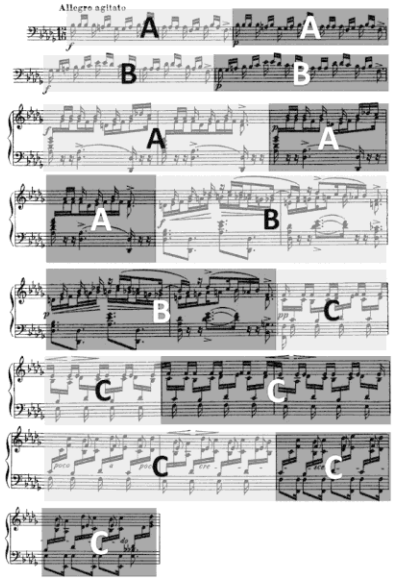
Tokenization

- Segment the data into a sequence of tokens
  - A token is one of the vocabulary formed by the tokenization methods
  - It is represented as an one-hot vector or an index number
- Language tokenization
  - Character → Sub-word → Word
    - A trade-off between vocabulary size and sequence length
    - Out-of-vocabulary issue
  - Byte-pair encoding (BPE) is used for the sub-word tokenization
- Symbolic music tokenization
  - MIDI-like, REMI, CPWord, Octuple, MMM, …
  - https://miditok.readthedocs.io/

Learning models

- Need to learn the structure of music
  - Melody and accompaniment
  - Repetition and variation (self similarity)
  - Consistent generation: note patterns or style
- Learning Models
  - RNN, VAE, Transformer
- Model evaluation
  - Objective: statistical metrics
  - Subjective: listening test





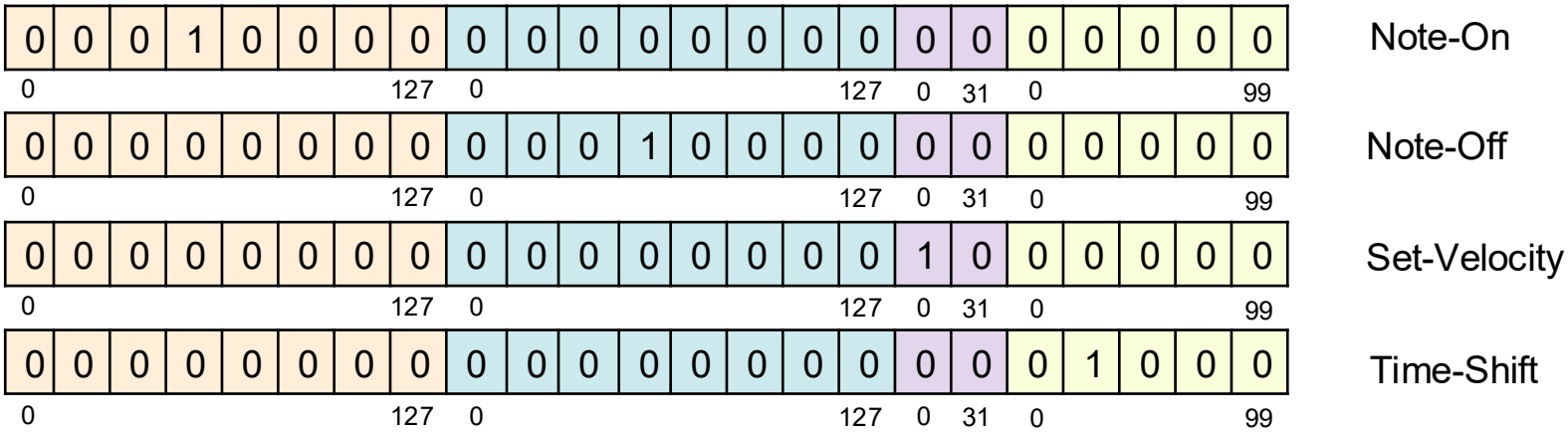
PerformanceRNN

- Dataset and goal
  - MAESTRO: performance piano MIDI files
  - Simultaneously composting and performing piano music
- Tokenization
  - “MIDI-like”
- Models
  - A simple auto-regressive RNN model

This Time with Feeling: Learning Expressive Musical Performance: Sageev Oore, Ian Simon, Sander Dieleman, Douglas Eck, Karen Simonyan, 2018

MIDI Tokenization

- Four types of MIDI-like event
  - Note-on (128 pitches), note-off (128 pitches), set-velocity (32 quantized velocities), time-shift (100 shifts: 10ms to 1000 ms)
  - 388-dim one-hot vector (388=128+128+32+100) → 388 vocabulary tokens



This Time with Feeling: Learning Expressive Musical Performance: Sageev Oore, Ian Simon, Sander Dieleman, Douglas Eck, Karen Simonyan, 2018

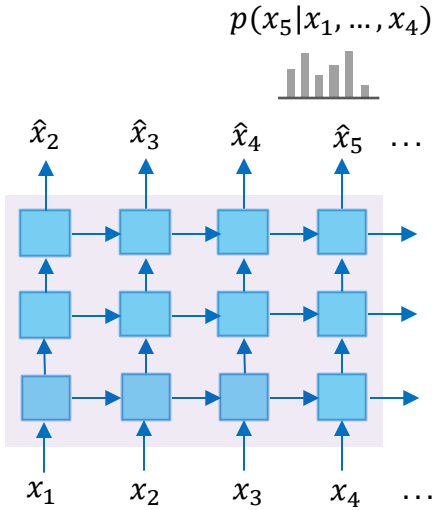
MIDI Tokenization

- The time shift event compresses sustained note states into a single event
  - A typical 30-sec clip might contain about 1200 event tokens

This Time with Feeling: Learning Expressive Musical Performance: Sageev Oore, Ian Simon, Sander Dieleman, Douglas Eck, Karen Simonyan, 2018

RNN-based Model: Training

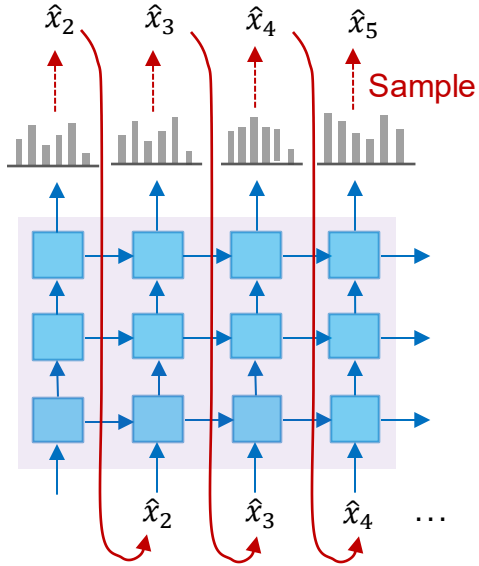
- Data augmentation
  - Tempo change and key transpose
- Three layers of LSTMs and the softmax output
  - One-hot MIDI-like event vector
  - The cross-entropy loss
  - Teacher-forcing: the ground output is used for input instead of the predicted output in the training phase



This Time with Feeling: Learning Expressive Musical Performance Sageev Oore, Ian Simon, Sander Dieleman, Douglas Eck, Karen Simonyan, 2018

RNN-based Model: Inference

- Generate the output using the trained model
  - Start from a random sample (unconditional) or an initial conditional input (“priming” or “continuation”)
  - Sample from the softmax output: multinomial distribution
  - The sampled output is used as input at the next step



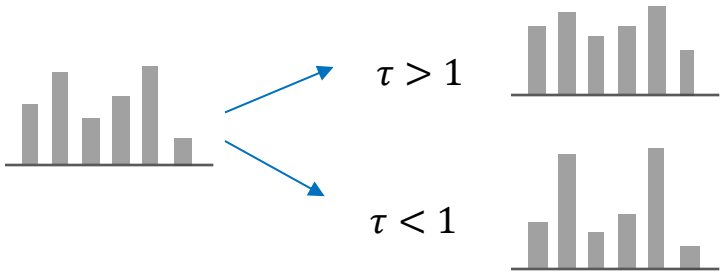
An intuitive animation about auto-regressive models: https://twitter.com/i/status/1327775912352493568

RNN-based Model: Inference

- The softmax temperature controls **musical diversity**
  - $\tau > 1 : P_t$ becomes more uniform
    - **More diverse** output are generated
  - $\tau < 1 : P_t$ becomes more spiky
    - **Less diverse** output are generated

$$P_t(w) = \frac{\exp(S_w/\tau)}{\sum_{w'} \exp(S_{w'}/\tau)}$$

Softmax output



RNN-based Model: Result

- Generation examples (unconditional generation)
  - https://magenta.tensorflow.org/performance-rnn

$\tau = 1$

$\tau = 0.9$

$\tau = 1.5$

- Issues
  - The result sounds natural in short terms but note patterns are not coherent and keeps diverging: the **long-term dependency** issue
  - Need better models capable of learning wider music context

Evaluating Music Language Model

- Objective evaluation
  - Perplexity (PPL): measure the likelihood of the generated output
    - Inverse probability of the corpus
    - Lower PPL is better

$$P(X) = \prod_{t=1}^{T} \left(\frac{1}{P_{LM}(x_t|x_{<t})}\right)^{1/T}$$

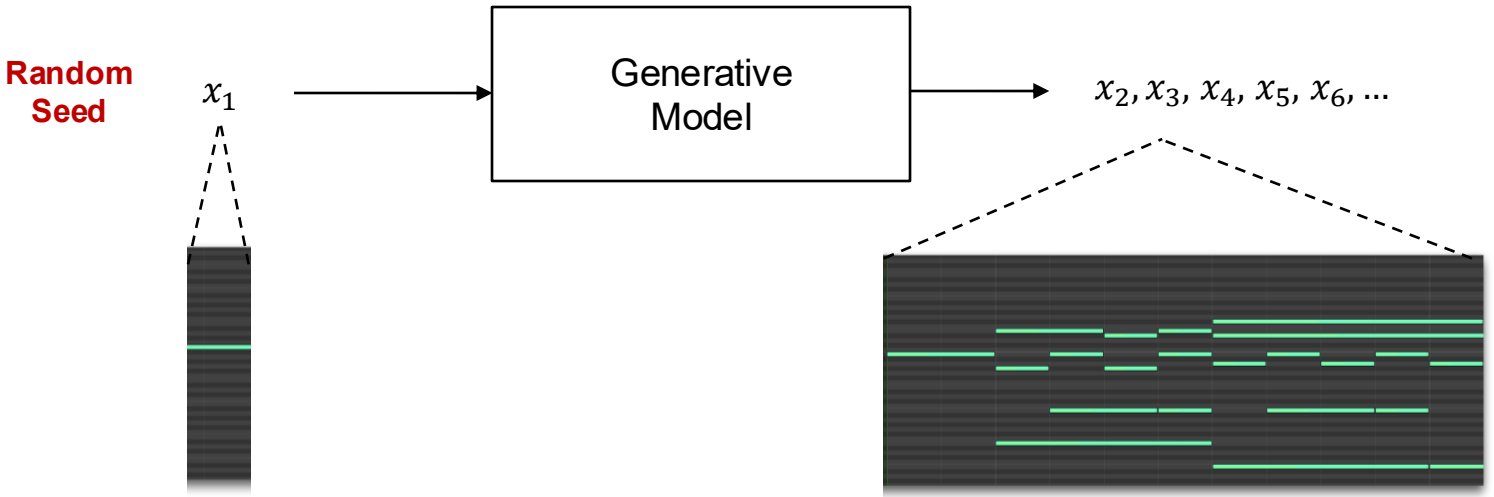
  - Comparing musical statistics
    - Pitch: pitch count, pitch class histogram, pitch transition histogram, pitch range
    - Rhythm: note count, average inter-onset-interval, note length histogram, note length transition histogram

- Subjective evaluation
  - Mean opinion score (MOS): scale from 1 to 5

On the evaluation of generative models in music, Li-Chia Yang, Alexander Lerch, Neural Computing and Applications, 2020

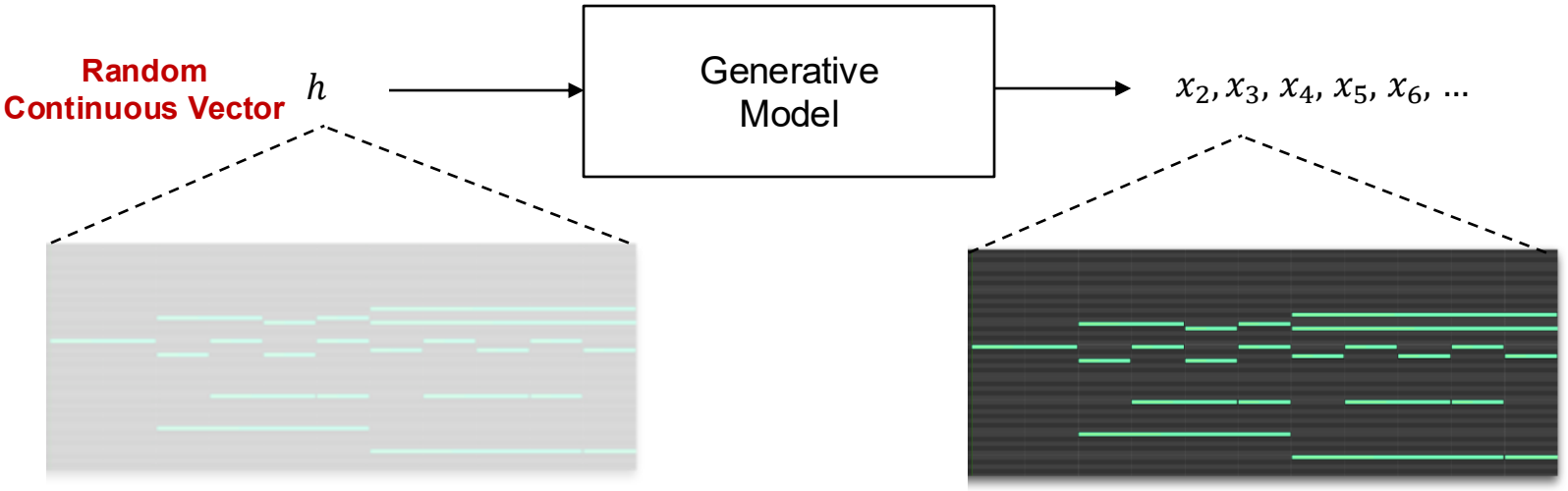
Issue in the RNN-based Model

- There is **a large contextual gap** between the input and the output
  - For different random inputs, the generated output sequences will be arbitrary



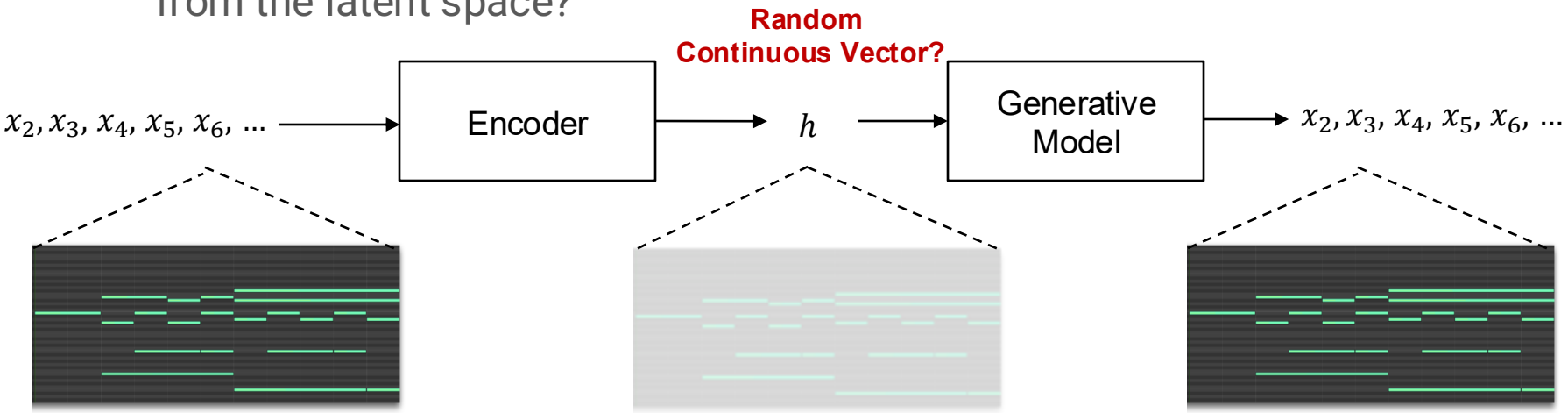
Issue in the RNN-based Model

- Can we feed a **random continuous vector** that governs the entire context of the output such that the input renders a smooth transition of generated sequences ?



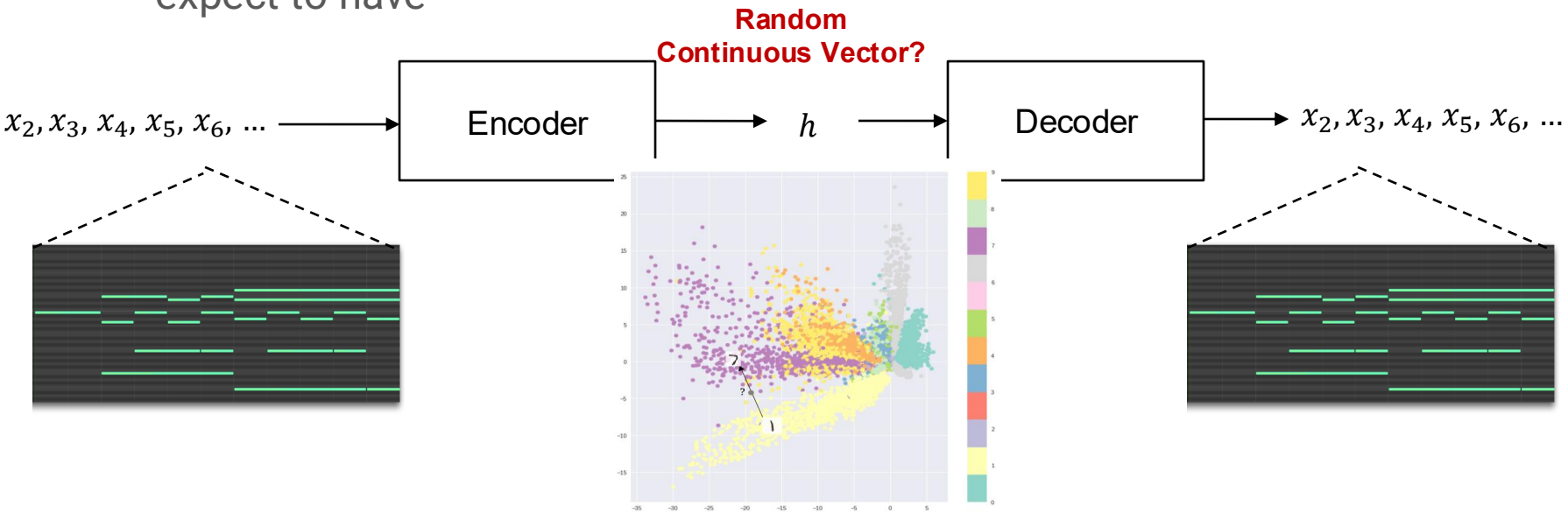
Auto-Encoder

- We can compress the context of the sequence into a single vector using the auto-encoder structure
  - But, can we sample a random vector and generate a meaningful sequence from the latent space?



Auto-Encoder

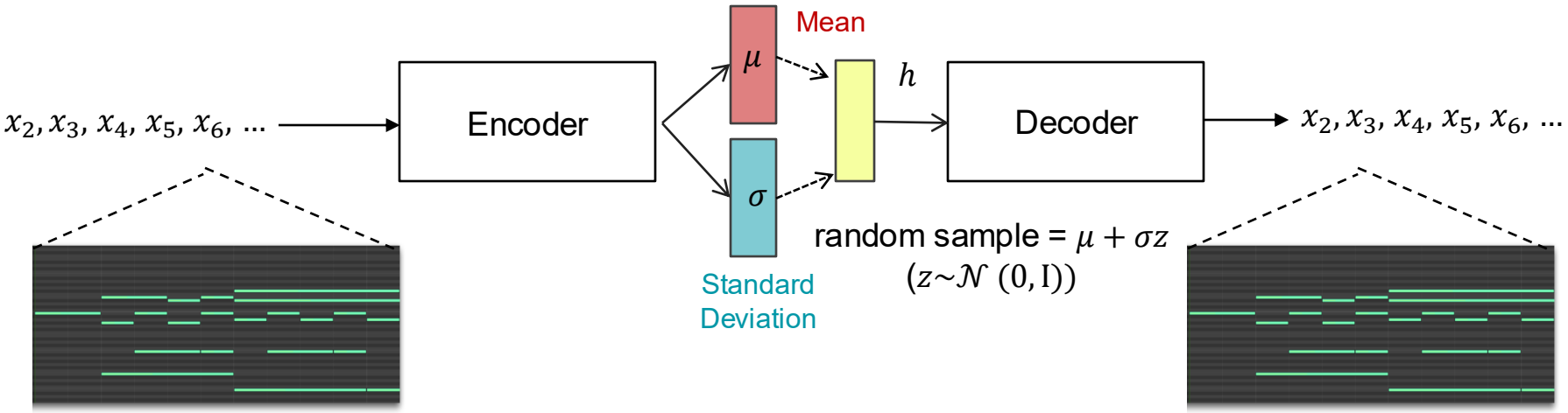
- The latent space may not be continuous
  - There are empty space between the latent vector clusters
  - The generated output from the empty space will be different from what you expect to have



Source: https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf

Variational Auto Encoder (VAE)

- Model the latent space using randomly sampled latent vectors with a probabilistic model such as Gaussian
  - The encoder yield two vectors for mean and standard deviation



Variational Auto Encoder (VAE)

- Optimize the network using the maximum likelihood estimation
  - The estimation is intractable and so an approximated method is used:
    - Maximize the lower bound of the log likelihood
  - This ends up with minimizing two terms: the **reconstruction error** and **KL divergence between the Gaussian distributions**

$$l(W;x) = \|x - \hat{x}\|^2 + KL(\mathcal{N}(\mu(x), \sigma(x)) \parallel \mathcal{N}(0, \mathrm{I}))$$

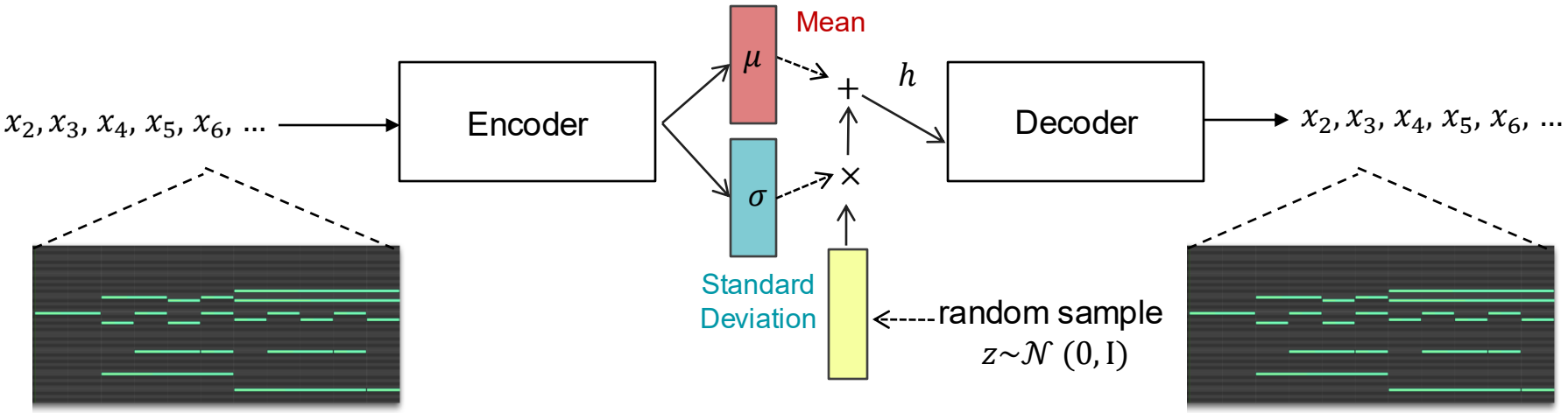
Reconstruction error

KL divergence: make the distribution of latent vectors have zero mean and unit variance

Auto-Encoding Variational Bayes, Diederik Kingma, Max Welling, 2014

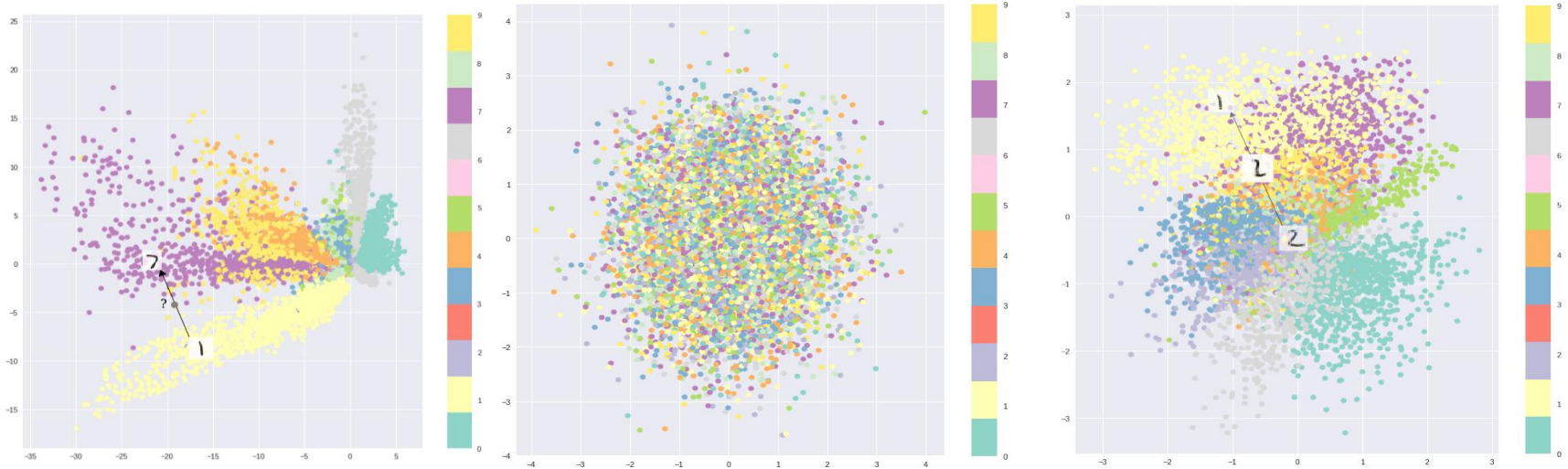
Variational Auto Encoder (VAE)

- Re-parameterization
  - Enables gradient flow by detouring the sampling process



Variational Auto Encoder (VAE)

- Distribution in the latent space
  - By using both KL divergence and reconstruction error, the space can be discriminative as well as continuous



Reconstruction only

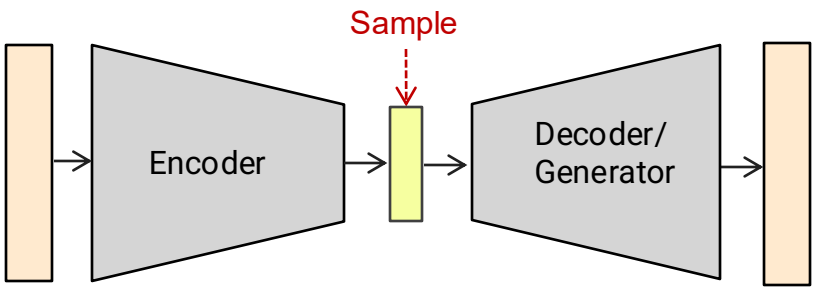
KL divergence only

Both

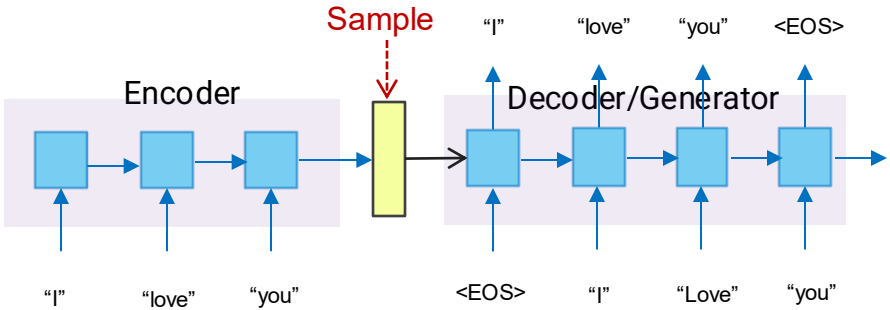
Source: https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf

Variational Auto Encoder (VAE)

- The encoder-decoder can be any neural network module
  - CNN: image audio
  - RNN: text, symbolic music



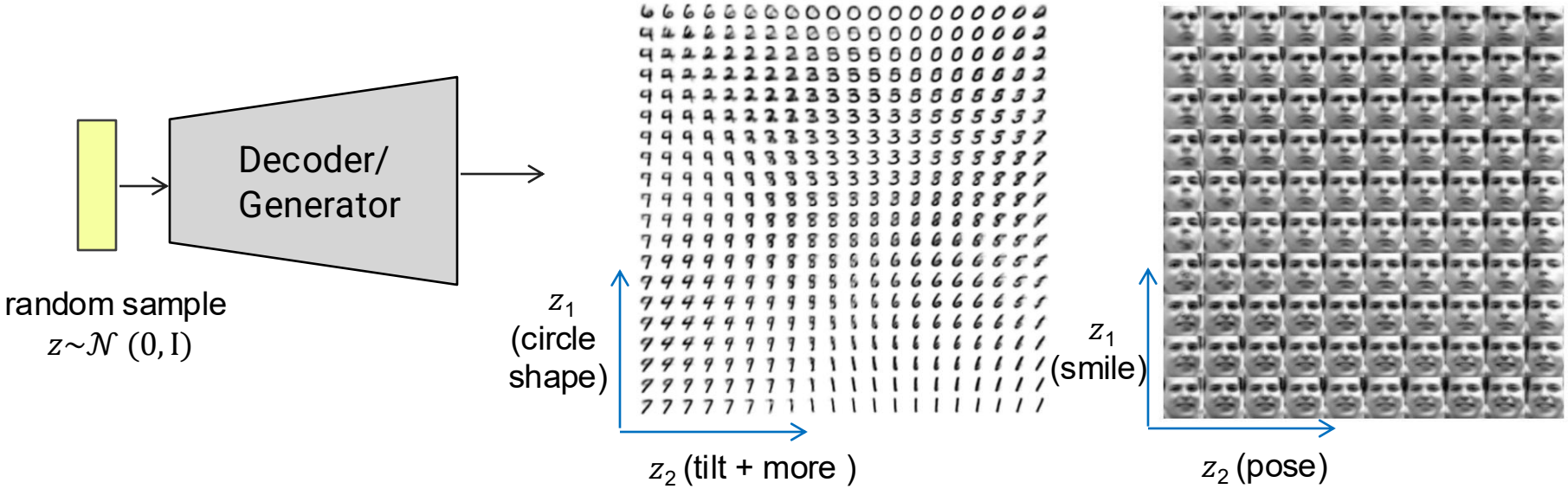
VAE with CNN



VAE with RNN

Variational Auto Encoder (VAE)

- Generate data by taking a random vector from the unit Gaussian
  - Data manifold are generated from varying z

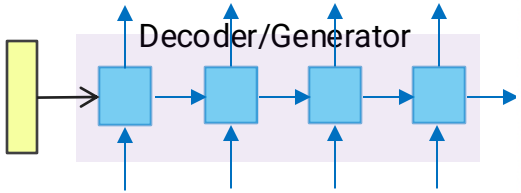


Generation from the 2-D latent space z

Auto-Encoding Variational Bayes, Diederik Kingma, Max Welling, 2014

Variational Auto Encoder (VAE)

- Generate data by taking a random vector from the unit Gaussian
  - Data manifold are generated from varying z



random sample
$z \sim \mathcal{N}(0, \mathrm{I})$

no .
he said .
“ no , ” he said .
“ no , ” i said .
“ i know , ” she said .
“ thank you , ” she said .
“ come with me , ” she said .
“ talk to me , ” she said .
“ do n’t worry about it , ” she said .

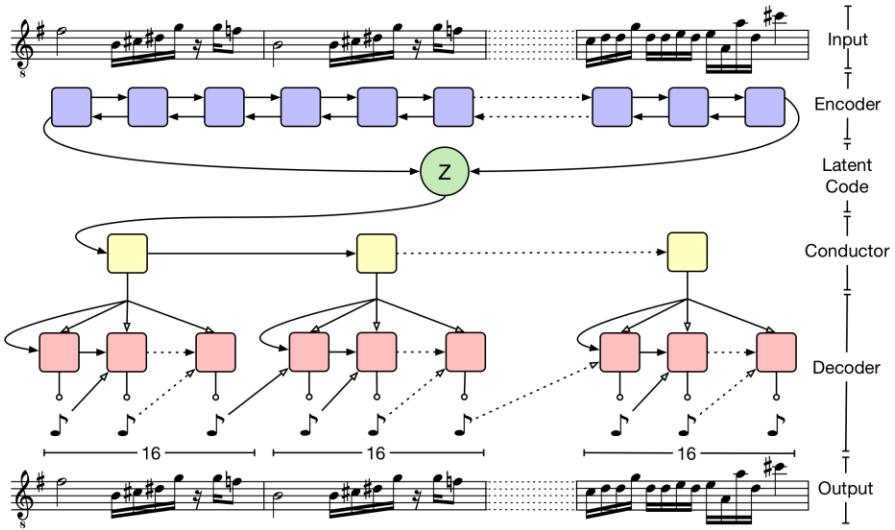
this was the only way .
it was the only way .
it was her turn to blink .
it was hard to tell .
it was time to move on .
he had to do it again .
they all looked at each other .
they all turned to look back .
they both turned to face him .
they both turned and walked away .

Interpolated sentences between pairs of random points in the latent space z

Generating Sentences from a Continuous Space, Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, Samy Bengio, 2016

MusicVAE

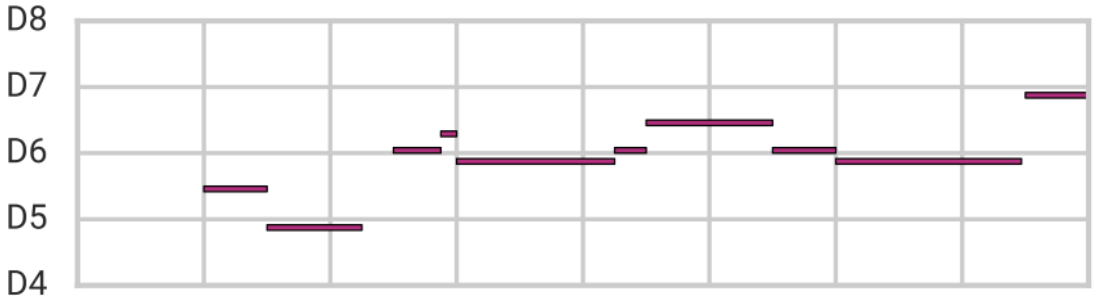
- Encoder: bidirectional RNN
  - The two hidden units at both ends are concatenated
  - The latent vector corresponds to a measure-level unit (2 bar ~ 16 bars)
- Decoder: hierarchical RNN
  - **Conductor RNN**: learns high-level dependency in the measure level
  - **Language model RNN**: condition from the conductor RNN is concatenated with the previous output as input at the next step



A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music, Adam Roberts, Jesse Engel, Colin Raffel, Curtis Hawthorne, Douglas Eck, 2018

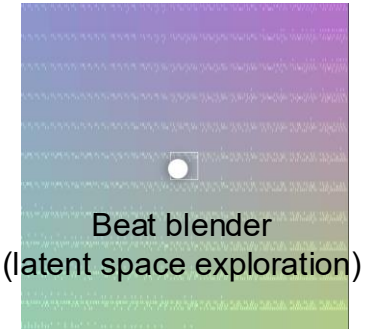
MusicVAE

- Training dataset
  - The Lakh MIDI dataset: Multi-track score MIDI
  - Use piano roll but quantized notes to 16th note events
  - Handle monophonic note sequences only (melody notes)
  - One event is 130 dimensional vector: note on (128 pitches), note off, rest
  - The input length of RNN (T) is 256 which corresponds to 16 measures (bars)



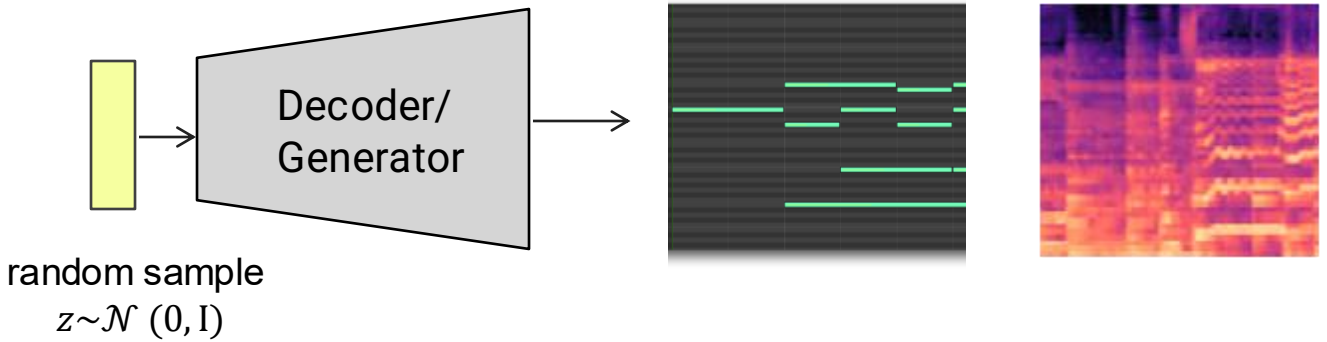
MusicVAE

- Generate a continuous transition of music generation on the latent space
  - **Beat-blender:** continuous move on the latent space to generate gradually changing music sequence
  - **Melody mix:** interpolation between two different melodies
- Demo
  - https://magenta.tensorflow.org/music-vae



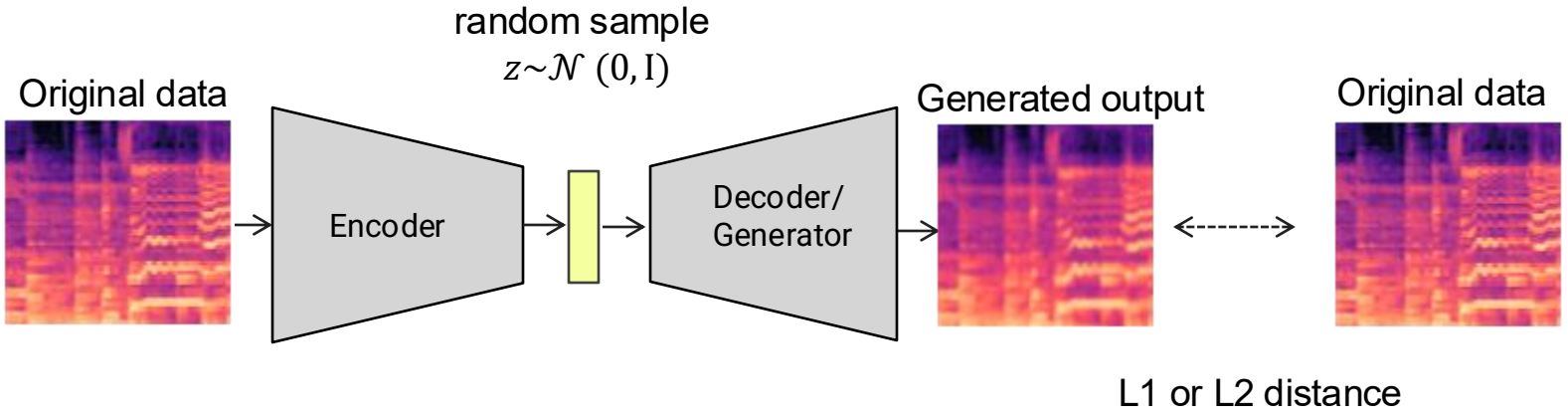
2D Models

- Generate music data as a 2D image
  - Symbolic domain: MIDI
  - Audio domain: spectrograms or other time-frequency representations
- Image generation models
  - VAE, GAN, Diffusion, …



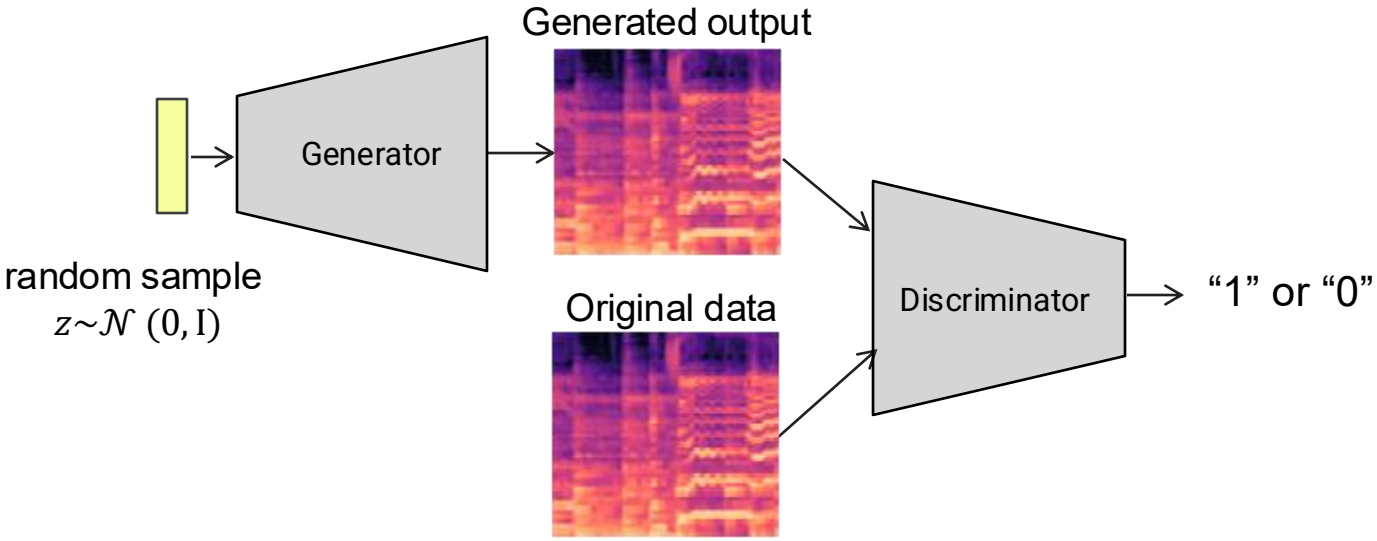
Variational Auto Encoder (VAE) for 2D Image Generation

- CNNs or Transformers are commonly used in this setting
  - Fast generation using parallel computing
  - The latent vector can be used to control the global structure
  - However, the generated result is often blurry (L1 or L2 distance)



Generative Adversarial Network (GAN)

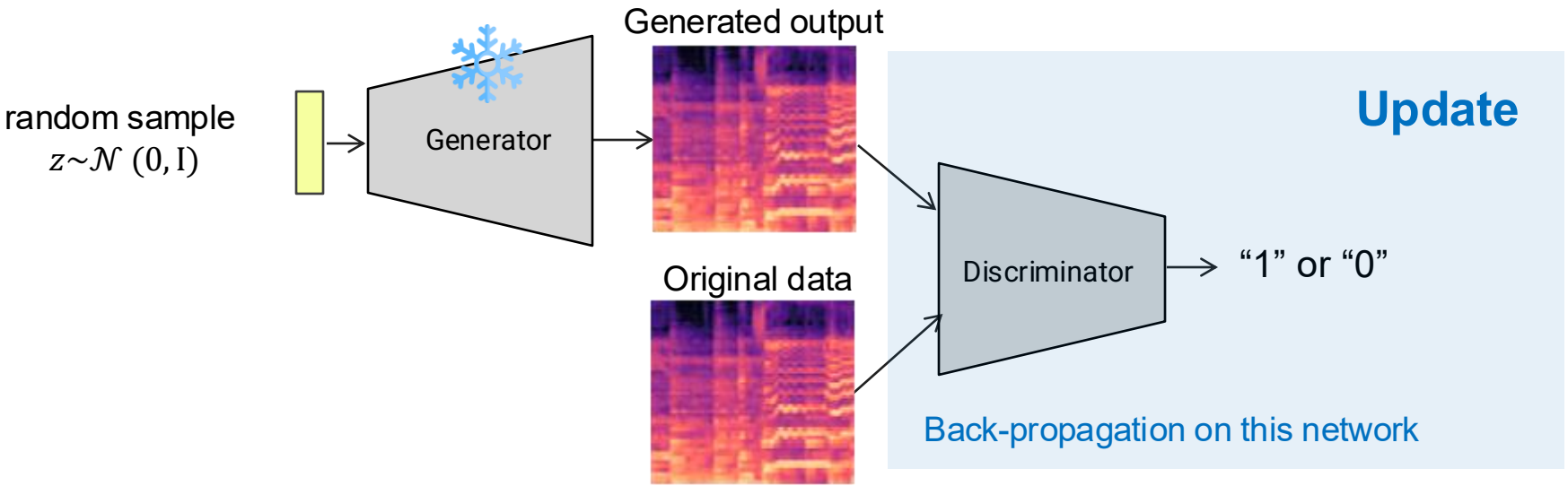
- Two-player game
  - Discriminator network: distinguish the generated output from the real ones
  - Generator network: fool the discriminator by generating the realistic output



Generative Adversarial Networks, Ian J. Goodfellow et al., NIPS, 2014

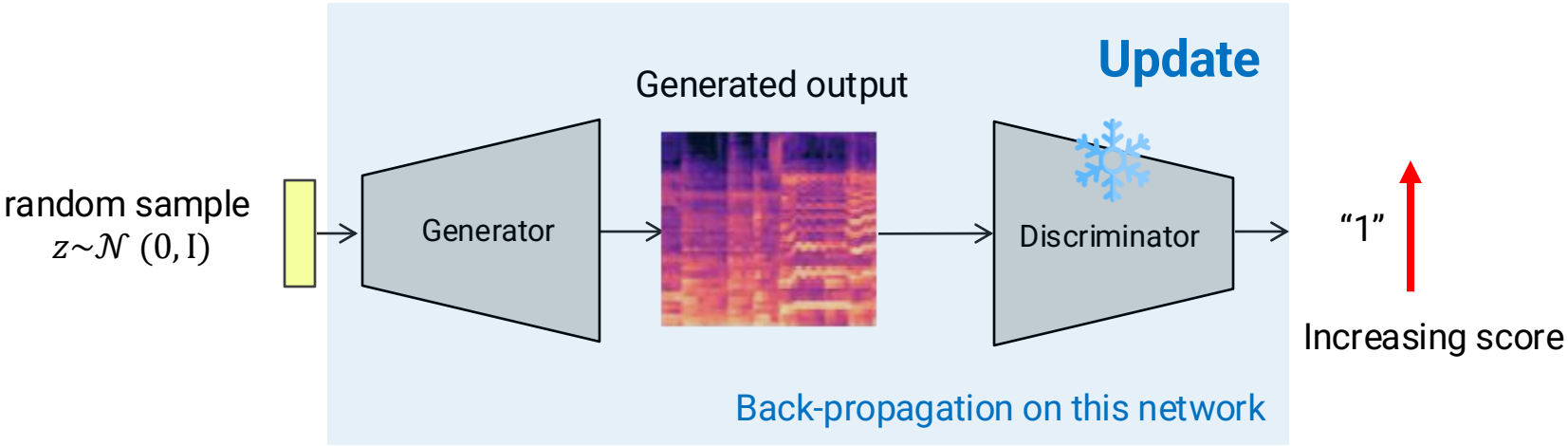
Training GAN

- Step #1: initialize the generator and discriminator networks
- Step #2: fix the generator network and generate the output
- Step #3: update the discriminator network as a binary classifier



Training GAN

- Step #4: fix the discriminator network and update the generator network
  - Try to fool the discriminator by increasing the output score (or generating real-looking images)
- Step #5: repeat step #2, #3, and #4 until convergence



Generative Adversarial Network (GAN)

- Minimize the minimax loss

$$\min_G \max_D E_{x \sim P_{data}(x)}[\log D(x)] + E_{z \sim P_{z(z)}}[\log(1 - D(G(x)))]$$

(Negative) logistic loss: Positive examples Negative examples

  - Discriminator: maximize the objective function such that $D(x)$ is close to 1 and $D(G(x))$ is close to 0
  - Generator: minimize the objective function such that $D(G(x))$ is close to 1

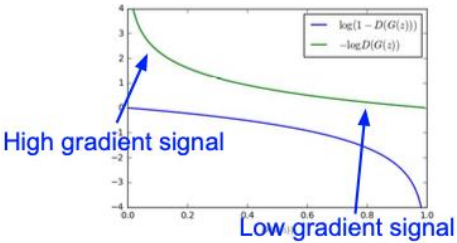
Generative Adversarial Network (GAN)

- Alternatively,
  - Discriminator: maximize the objective function such that $D(x)$ is close to 1 and $D\big(G(x)\big)$ is close to 0
    - Gradient ascent on the discriminator

$$\max_D E_{x\sim P_{data}(x)}[\log D(x)] + E_{z\sim P_{z(z)}}\big[\log(1 - D\big(G(x)\big))\big]$$

  - Generator: maximum the objective function such that $D\big(G(x)\big)$ is close to 1
    - Gradient ascent on the generator
    - This alternative objective is more easily trained

$$\min_G E_{z\sim P_{z(z)}}\big[\log(1 - D\big(G(x)\big))\big] \longrightarrow \max_G E_{z\sim P_{z(z)}}\big[\log D\big(G(x)\big)\big]$$



[Stanford CS231n]

Issue in Training GAN

- The discriminator provides the generator with gradients as a guidance for improvement
  - Discrimination is easier than generation
  - Discriminator tends to provide large gradients
  - Result in unstable training of the generator
- There are alternatives of the original minimax loss
  - Wasserstein loss: critic instead of discriminator
  - Boundary Equilibrium GAN (BEGAN): fast and stable convergence
- Readings
  - https://lilianweng.github.io/posts/2017-08-20-gan/
  - https://arxiv.org/abs/2001.06937

GAN vs VAE

- When a model does not have enough capacity to capture all the variability in the data, different compromises can be made
  - GAN has the mode-seeking nature: causes mode collapse or mode missing
  - VAE has the mode-covering nature: causes blurred output

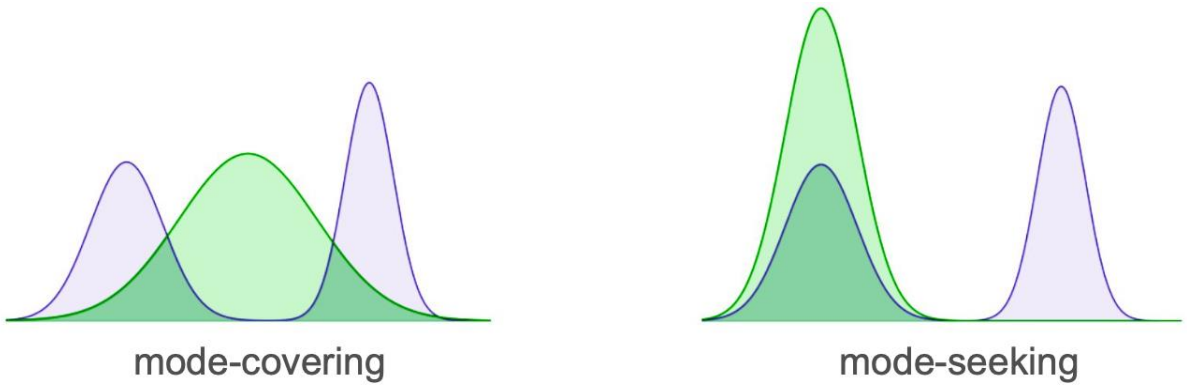
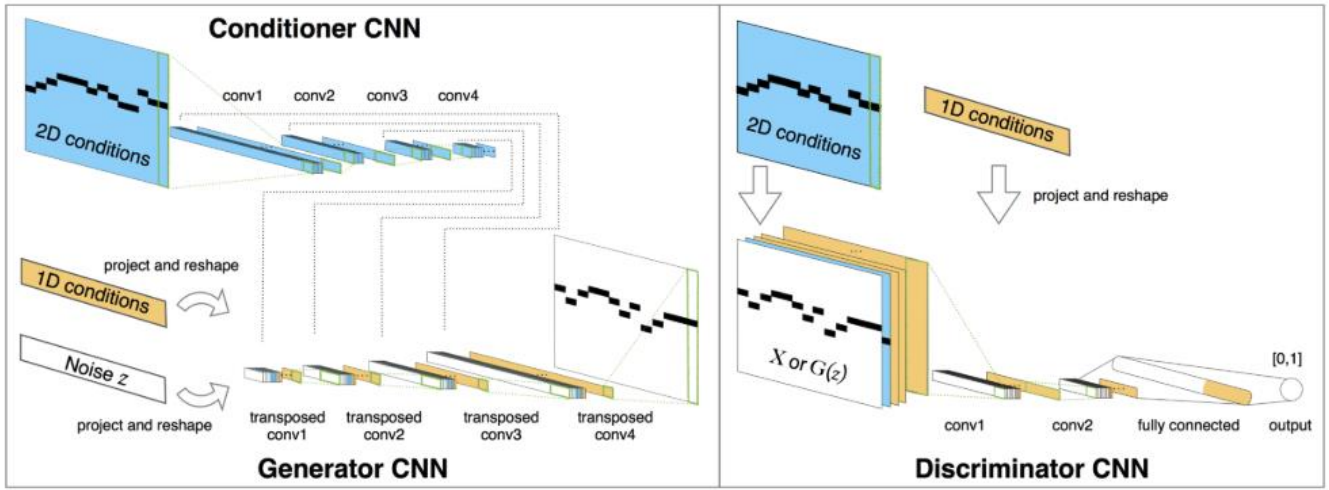


Image source: Sander Dieleman: https://sander.ai/2020/03/24/audio-generation.html

MIDINet

- Convolutional GAN for one-bar melody generation
  - Generate a piano-roll matrix
  - Conditioned on the previous bar or on the chord

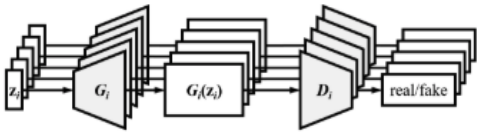


Demo page: https://richardyang40148.github.io/TheBlog/midinet_arxiv_demo.html

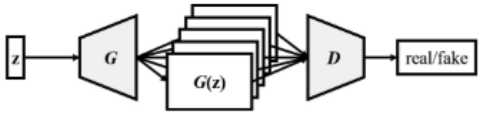
MidiNet: A Convolutional Generative Adversarial Network for Symbolic-domain Music Generation, Yang et al., ISMIR, 2017

MuseGAN

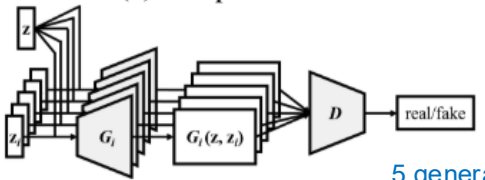
- Convolutional GAN for multi-track MIDI generation
  - 5 tracks (bass, drum, guitar, piano, strings& others) and 4 bars
  - Learn cross-track and cross-bar dependency



(a) Jamming model

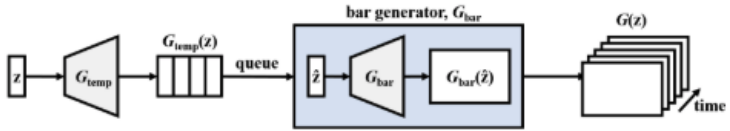


(b) Composer model

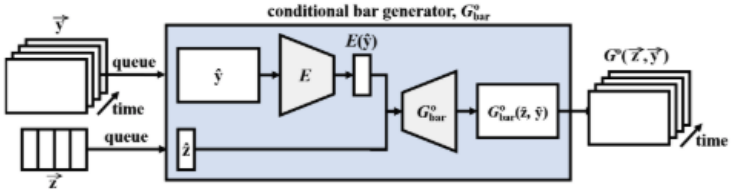


(c) Hybrid model

5 generators
& 1 discriminator



(a) Generation from scratch

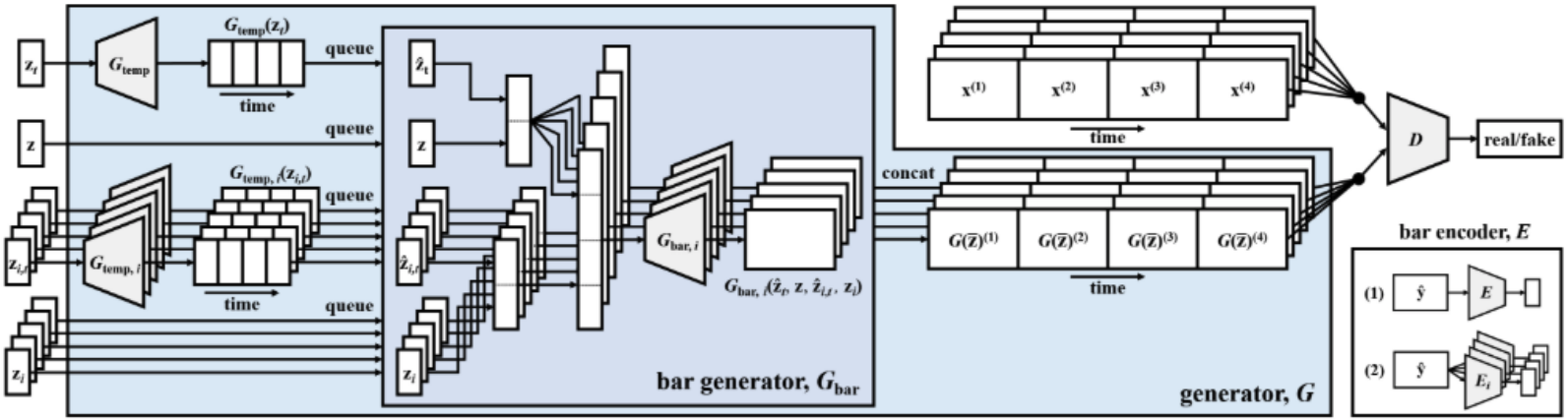


(b) Track-conditional generation

MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment, Dong et al., AAAI 2018

MuseGAN

- Convolutional GAN for multi-track MIDI generation
  - 5 tracks (bass, drum, guitar, piano, strings& others) and 4 bars
  - Learn cross-track and cross-bar dependency



Demo page: https://hermandong.com/musegan/

MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment, Dong et al., AAAI 2018